A Novel Method to Control the Diversity in Cluster Ensembles

Milton Pividori[1,2], Georgina Stegmayer[1,2], and Diego Milone[2]

[1] CIDISI - Universidad Tecnológica Nacional - Facultad Regional Santa Fe
[2] sinc(i) - Universidad Nacional del Litoral - Facultad de Ingeniería y Ciencias Hídricas

Abstract. Clustering is fundamental to understand the structure of data. In the past decade the cluster ensemble problem has been introduced, which combines a set of partitions (an ensemble) of the data to obtain a single consensus solution that outperforms all the ensemble members. Although disagreement among ensemble partitions (diversity) has been found to be fundamental for success, the literature has arrived to confusing conclusions: some authors suggest that high diversity is beneficial for the final performance, whereas others have indicated that medium is better. While there are several options to measure the diversity, there is no method to control it. This paper introduces a new ensemble generation strategy and a method to smoothly change the ensemble diversity. Experimental results on three datasets suggest that this is an important step towards a more systematic approach to analyze the impact of the ensemble diversity on the overall consensus performance.

Keywords: consensus clustering, ensemble diversity, cluster ensemble generation

1 Introduction

Clustering is fundamental to understand the structure of a dataset [2]. It has been used in a wide range of areas, including physics, engineering, medical sciences, social sciences and economics. Clustering algorithms partition data into groups called clusters, in such a way that data objects inside the same cluster are more similar than those in different ones [20]. The output of these techniques is called partition. The correct choice of a clustering algorithm, or even the setting of its parameters, requires the user to have at least some knowledge about the dataset, which data distribution the algorithms assumes and how its parameters setting could affect the final result [11]. In fact, clustering algorithms are developed to solve a wide range of different problems. There is no universal technique to solve all of them. Different and equally valid solutions can be obtained from different algorithms. That is one of the reasons why clustering is accepted in the community as an ill-posed problem [11, 12, 21, 22]. Therefore, the inexperienced user runs the risk of picking an inappropriate algorithm, or even a proper one with a wrong set of parameters. While all these issues are some of the main motivations behind cluster ensembles [10, 19], another interesting applications include the possibility to reuse the current knowledge about the data and perform distributed data mining [18], where different partitions of the data are present in geographically distributed locations.

In the past decade, *cluster ensembles* have emerged as an important approach to combine a set of partitions of the data, called ensemble, into one consolidated solution that has an improved overall accuracy [14, 18, 19]. Given the ill-posed nature of clustering, accuracy or performance is typically measured by comparing the final solution against a known reference partition, generally based on the class labels that come with the dataset used [10, 15, 22]. Although this reference partition may not be the only valid structure of the data [7], many studies have tried to find how ensembles should be built or which characteristics they should have to obtain a high performance. Namely, the level of disagreement between ensemble members, which is called *ensemble diversity*, has been identified as a fundamental factor for success [7, 18], and many diversity measures have been proposed [3, 14].

In the literature, different opinions can be found when analyzing the relationship between ensemble diversity and performance. Some studies suggest that more diverse ensembles are better to get more accurate solutions [3, 10], while others, in contrast, have proposed that a medium diversity is the preferred choice [7]. In addition to these contradictory statements, a high variability has been found when a proposed approach is used not only from one dataset to another, but also when different ensemble generation strategies are employed. The diversity vs accuracy plots also reveal that ensembles with similar diversities can have very different accuracies. When this is observed, two possible explanations can be formulated as hypothesis: 1) while one *type of diversity* is being measured and changed, another *hidden types*, not measured, are changing as well, thus leading to confusing results; 2) it is difficult to precisely generate ensembles with different diversity values, which could cause a biased analysis.

The previous facts lead us to propose a method to obtain a fine-grained control of an ensemble diversity. For this purpose, a new strategy to generate ensembles is introduced, along with a method to smoothly change the diversity of an ensemble. Results show that this method is able to precisely generate ensembles with different diversities, representing a first step towards a more systematic approach to analyze the impact of diversity on the final solution.

This paper is organized as follows. Section 2 describes the cluster ensemble problem and a diversity measure. In Section 3, a new strategy to generate ensembles is introduced, in addition to a novel method to smoothly change the diversity of an ensemble. Section 4 describes the evaluation procedure and the results found, while Section 5 summarizes the conclusions, possible improvements and future work.

2 The Cluster Ensemble Problem

The *cluster ensemble problem* was firstly defined a decade ago and several extensions have been presented since then. It consists in combining a set of partitions to obtain a single consolidated one without accessing the data features or the algorithms that generated that set of partitions [18]. In this section, a *cluster ensemble framework* is described, along with a fundamental factor for its success that has been studied in the literature: the *ensemble diversity*.

2.1 A Cluster Ensemble Framework

A *cluster ensemble framework* for knowledge reuse was initially introduced in [18], and it is depicted in Figure 1. The data, shown at the left, is processed by clustering algorithms, which are called *clusterers* and denoted as Φ_i. As an example,

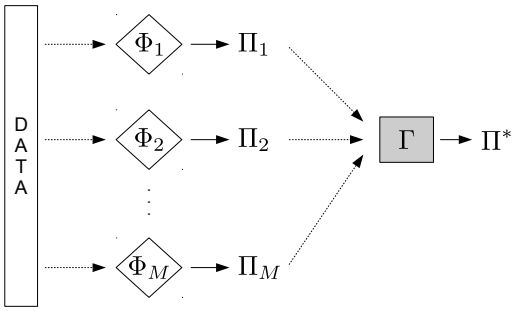


Fig. 1. The cluster ensemble framework and its components.

three different clusterers could be k-means [8] with $k = 5$, k-means with $k = 3$, or a SOM [13] with a map size of 4×4. Each clusterer produces one *partition* of the data, Π_i. There are M clusterers in the framework, thus M partitions are generated. The set of partitions $\Pi_1, \ldots, \Pi_M$ produced by the clusterers is called *ensemble*. The ensemble is the input of the next component, the *consensus function* Γ, which produces a single consolidated partition Π^*, called *consensus partition*.

The objective of Γ is to maximize the information shared between the consensus partition and the ensemble members. To measure the information shared between two partitions, the Normalized Mutual Information (NMI) has been proposed [5]

$$\Upsilon(\Pi_i, \Pi_j) = \frac{I(\Pi_i, \Pi_j)}{\sqrt{H(\Pi_i)H(\Pi_j)}}, \tag{1}$$

where $I(\Pi_i, \Pi_j)$ represents the mutual information between partitions Π_i and Π_j, while $H(\cdot)$ is the partition entropy. NMI is a symmetric measure and ranges from 0 to 1.

To quantify the information shared between a single partition $\Pi^{'}$ and a set of partitions $\Lambda = \Pi_1, \ldots, \Pi_M$, the Average Normalized Mutual Information (ANMI) is defined as

$$\bar{\Upsilon}(\Lambda, \Pi^{'}) = \frac{1}{M} \sum_{i=1}^{M} \Upsilon(\Pi^{'}, \Pi_i). \tag{2}$$

Therefore, the objective function for Γ can be formally defined and consists in deriving a consensus partition Π^* that maximizes the ANMI:

$$\Pi^* = \arg\max_{\Pi^{'}} \sum_{i=1}^{M} \Upsilon(\Pi^{'}, \Pi_i). \tag{3}$$

Several consensus functions have been proposed to solve the cluster ensemble problem. They use different approaches to combine an ensemble into a single consensus partition, for instance: 1) by using graph theory, which employs graph representations and partitioning algorithms [18]; 2) by using the cluster labels as features and clustering them [19]; 3) by relabeling the clustering results to minimize their disagreement [6]; 4) by using the link analysis methodology to

find the similarities between clusters [10]; 5) by using a pairwise similarity matrix for data objects and a similarity-based clustering algorithm over it [3, 7]. Among the graph-based approaches, a well-known consensus function is the Meta-CLustering Algorithm (MCLA), which will be used in this paper.

It has been been stated that the consensus partition Π^* has better average performance than all the individual partitions in the ensemble [3, 19]. The performance or accuracy typically refers to the degree of similarity between the consensus partition and a known reference partition, which can be calculated using (1). Although the objective of the consensus function consists in maximizing the information shared, some studies evaluate it by using the accuracy [1, 10].

2.2 Ensemble Diversity

Diversity among a pair of partitions can be defined as a measure that quantifies the degree of disagreement between them. A simple diversity measure consists in calculating the complement of a similarity measure [3], like $D(\Pi_i, \Pi_j) = 1 - \Upsilon(\Pi_i, \Pi_j)$. *Ensemble diversity*, on the other hand, refers to the level of disagreement among ensemble members.

Two main approaches have been proposed to measure ensemble diversity [7]: pairwise and non-pairwise. In the former, every partition member of the ensemble is compared to the rest. In the latter, a consensus partition is first derived from the ensemble and every partition member is then compared with it. The NMI or the Adjusted Rand Index (ARI) [9] are generally used as the indices to compare a pair of partitions [3, 10]. A pairwise measure based on NMI is defined as

$$D_p(\Lambda) = \frac{2}{M(M-1)} \sum_{i=1}^{M-1} \sum_{j=i+1}^{M} (1 - \Upsilon(\Pi_i, \Pi_j)), \tag{4}$$

where the p subindex stands for a pairwise approach.

Several approaches exist to generate diversity in ensembles: by using different clustering algorithms [18], varying their parameters [7, 10], projecting data into different subspaces [3, 21], using different features of the dataset [18], based on bagging and boosting [16, 17] or a combination of them [23]. These approaches consist in randomly generate a set of ensembles by following a generation strategy and hoping to obtain a wide range of diversities. However, there is no method to generate an ensemble with a determined diversity.

3 A Method to Control the Diversity

The cluster ensemble framework, depicted in Figure 1, suggests the need of two main activities: 1) the ensemble generation, and 2) the application of a consensus function to obtain a consensus partition, which can be seen as the final result of the whole process. The main contribution of this article is focused on the first activity, the ensemble generation. In this section, a new approach to generate an ensemble is proposed, along with a method to smoothly change its diversity. This fine-grained method to control the diversity represents a step towards a more effective way to study the impact of the diversity on the quality of the final consensus partition.

First, a strategy to generate an ensemble based on groups of partitions is described. After that, a novel process to make small changes in the original ensemble to obtain a fine-grained control of diversity is proposed. This process heavily depends on the generation strategy, and consists in taking its output and

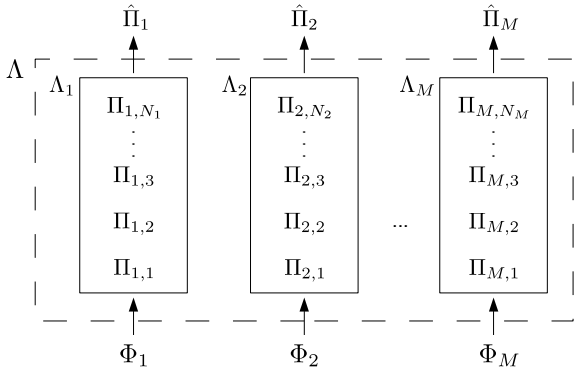


Fig. 2. Generation of an ensemble based on groups of partitions, and definition of a representative partition for each one. The solid line indicates each group boundary, and the dashed one the whole ensemble.

produce a set of modified ensembles. By finding the relationship of each group with the rest of the original ensemble, the method is able to produce a smooth change in the diversity of the output ensembles.

3.1 Ensemble Generation Using Groups of Partitions

While there are several methods to create an ensemble, a common approach involves the process of creating different partitions of the data by using a fixed algorithm and randomly varying some of its parameters. For this purpose, k-means is generally used and some of the following schemes for selecting the number of clusters is employed: k is fixed and the cluster centers are randomly initialized; k is chosen randomly within an interval $[k_{min}, k_{max}]$ [4, 7, 14].

The proposed method uses a combination of both approaches: rather than using a fixed k for all the ensemble members, an interval of k values is determined and used. For each k value in this interval, the corresponding clusterer is run a number of times with random initializations, producing a *group of partitions*. Therefore, the ensemble is composed by these groups of partitions, and there are as many groups as k values in the interval. A diagram of this ensemble generation process is shown in Figure 2. Each group is composed by partitions $\Pi_{i,j}$ generated by the same clusterer $\Phi_i \rightarrow \Lambda_i = \{\Pi_{i,1}, \Pi_{i,2}, \ldots, \Pi_{i,N_i}\}$. All these groups of partitions form the ensemble, and there are as many groups as k values: $\Lambda = \{\Lambda_1, \Lambda_2, \ldots, \Lambda_M\}$. Although each group can have different sizes N_i, the original ensemble has groups with a fixed size.

The ensembles generated by this method have a known structure: groups of partitions where each partition within a group is generated by the same clusterer, only varying from the other group members in the initialization of the cluster centers (which is random). Decoupling this definition from centroid-based algorithms and thus making it more generic, each group member differs from the rest of the group only by one random component of the clustering algorithm used for its creation.

3.2 Representative Partitions for Group Comparison

The generation strategy described in the previous section should provide some benefit to achieve our final goal of controlling the ensemble diversity in a smoothly

manner. As the ensemble structure is already known, it is possible to take advantage of it by analyzing the relationship between each group. A naive way to get this information would be to generate all possible combinations of groups taken by 2 and compare all the members of the first group against members of the second one. While it seems correct, this calculation could be very computational intensive, and the number of groups and their sizes would be an important limit when generating ensembles.

To reduce the computation complexity when comparing groups, a different approach is proposed here. Note that it is known that all partitions within a group share the same clusterer. Therefore, each group itself could be considered as a kind of cluster inside the whole ensemble. Partitions within each group should be similar, or at least it can be assumed that they share some structure. Then, instead of the naive comparison described above, it could be possible to take advantage of the ensemble structure and obtain a representative partition for each group in the ensemble. This representative partition (also referred just as the *representative*) should be the single partition that best represents the complete group from where it was taken or derived. Assuming that good representatives can be obtained, then a simple comparison between them would provide the comparative information needed about groups, without excessive computation. A convenient definition for the representative partition could be the one that maximizes the mutual information among group members, similar to (3). So its objective function is defined as

$$\hat{\Pi}_i = \arg\max_{\hat{\Pi}} \bar{\Upsilon}(\Lambda_i, \hat{\Pi}). \tag{5}$$

As each group of partitions was generated with a clusterer using the same $k = i$, a group Λ_i and its representative $\hat{\Pi}_i$ share the same subindex, which identifies the clusterer configuration.

The similarity between (3) and (5) suggests that any existing consensus function could be used as a method to get a representative partition. An alternative approach could be to look among the group members for a representative partition, thus the group member with the highest ANMI is chosen as the representative for its group. It is worth mentioning that, while the first approach would produce a completely new partition, different to all group members, the second one always chooses an existing one.

3.3 Ensemble Diversity Control

Once the representative partitions are obtained, a comparison between them could provide useful information about the groups of partitions. This information can be used to modify the original ensemble, thus obtaining a diversity change. That is achieved by changing the group sizes in the new ensemble, according to their relationship with the other groups.

For the comparisons, a *relationship matrix* R can be computed. For example, if the measure chosen is $R_s = \Upsilon(\hat{\Pi}_i, \hat{\Pi}_j)$, then a *similarity matrix* is obtained. On the other hand, a *dissimilarity matrix* can be generated if the measure is $R_d = 1 - \Upsilon(\hat{\Pi}_i, \hat{\Pi}_j)$. As an example, Figure 3 shows the R_s for the Iris dataset using k-means. By looking at the averages of the columns, $\bar{r}_i$, it can be seen that the group using $k = 2$ is the most different to the rest, with an average NMI of 0.59.

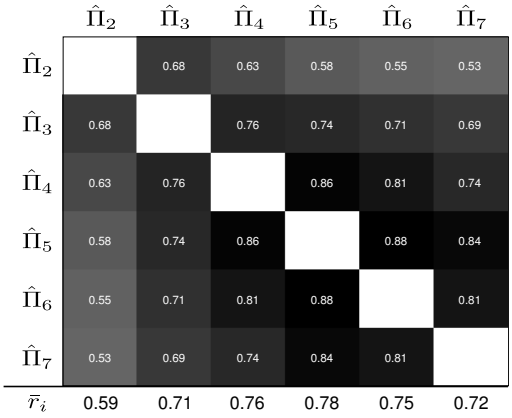


Fig. 3. Similarity matrix generated for the Iris dataset using k-means with $k \in [2, 7]$. The last row represents the column average (discarding the main diagonal).

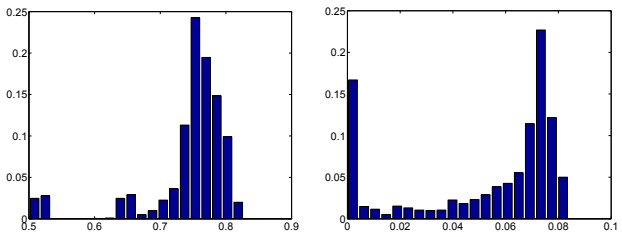
It is interesting to note that the structure of the ensemble allows to play with the groups proportion (number of members in each group) while still preserving most of the original ensemble. If a new ensemble is created from the original one and the number of members of the most diverse group is decreased at some factor, what would be the change in its diversity? By making small changes in the groups according to the information gathered in the relationship matrix R, it could be possible to explore its effects on the final diversity. An intuitive idea is that, if small decreases in diversity are desired, a possible action can be to slightly reduce the size of the most diverse groups and increase the proportion of the most similar ones. The opposite operation would produce diversity increases instead. Both mechanisms provide a diversity control method, and they are the main proposal of this work.

The averages $\bar{r}_i$ can serve as a guide to change the group proportions and obtain the desired diversity change through *representative weights*, $w_i \triangleq \bar{r}_i$. These weights will be used to graduate the proportion of the groups in the new ensemble $\tilde{N}_i = w_i |\Lambda| / \sum_i w_i$. Once the new sizes are estimated, the groups in the new ensemble are made up by uniformly sampling from the original group members. The new groups can have repeated partitions, as the new size could be larger than the number of available partitions for that group.

However, this procedure, which uses plain averages, generates only one new ensemble with a different diversity. This is far from what we have defined as a diversity control method. A fine-grained method should be able to generate ensembles according to a desired *level of diversity*. To achieve this, a function to gradually *emphasize* the differences between the values of $\bar{r}_i$ can provide such diversity control method. We propose to use the sigmoid function

$$s(\bar{r}_i, h) = \frac{1}{1 + e^{-h(\bar{r}_i - \bar{r})}}, \tag{6}$$

where parameter h controls its shape. When h approaches 0, the function turns into a linear weighting. Larger values for h, however, change its behavior into a step function. If small values for h are used, the new groups will have similar



(a) $w_i = \bar{r}_i$. (b) $w_i^h = s(\bar{r}_i, h)$ with $h = 150$.

Fig. 4. Histograms of (a) the similarity matrix averages and (b) the final weights for representatives after applying a sigmoid function. The Iris dataset was used.

sizes, and there will be almost no change from the original ensemble to the new one. When larger values for h are employed, larger differences in the new group sizes will be observed, causing the new ensembles to smoothly differ from the original.

The sigmoid function seems to be a good option to change the averages $\bar{r}_i$ into differently contrasted weights, according to the desired diversity change. This transformation is depicted in Figure 4, where two histograms are shown: a) the distribution of the averages $w_i = \bar{r}_i$ and b) how it is changed when the final weights are calculated after applying $w_i^h = s(\bar{r}_i, h)$. Clearly, the differences are more sharply contrasted.

Once the final weights w_i^h are calculated using the sigmoid function with parameter h, the sizes for each group are obtained using $\tilde{N}_i = w_i^h |\Lambda| / \sum_i w_i^h$, and their members are chosen by sampling from the original group members. Increasing values for h result in smooth changes in the ensemble diversities.

Finally, it is worth mentioning something more about the relationship matrix and how it affects the results. As it was previously said, each time h is increased, w_i values will be more sharply contrasted. If R is a similarity matrix, this means that the groups more similar to the rest of the ensemble will be privileged (obtaining larger weights), while the more diverse will be reduced or even completely discarded, as suggested by Figure 4(b). This results in a decrease of the new ensemble diversity. The opposite effect can be achieved if a dissimilarity matrix is used instead, that is to say, the diversity will be increased.

4 Results and Discussion

In this section, the proposed diversity control method along with the ensemble generation strategy were evaluated in different test cases. Three well-known datasets were used: 1) Iris[3], real dataset, 150 data objects, 3 classes; 2) Difficult Doughnut, artificial dataset, 500 data objects, 2 classes; 3) Four Gaussian, artificial dataset, 100 data objects, 4 classes. All classes are equally distributed.

4.1 Representative Partitions

It is important for this study to assess the quality of the representative partition, that is to say, its representativeness for the group. It is possible to exactly measure this quality according to the objective function defined in (5). Two

[3] `http://archive.ics.uci.edu/ml/datasets/Iris`

Table 1. Representative quality test for two methods, MCLA and Maximum ANMI. The Four Gaussian dataset was used with group sizes equal to 20.

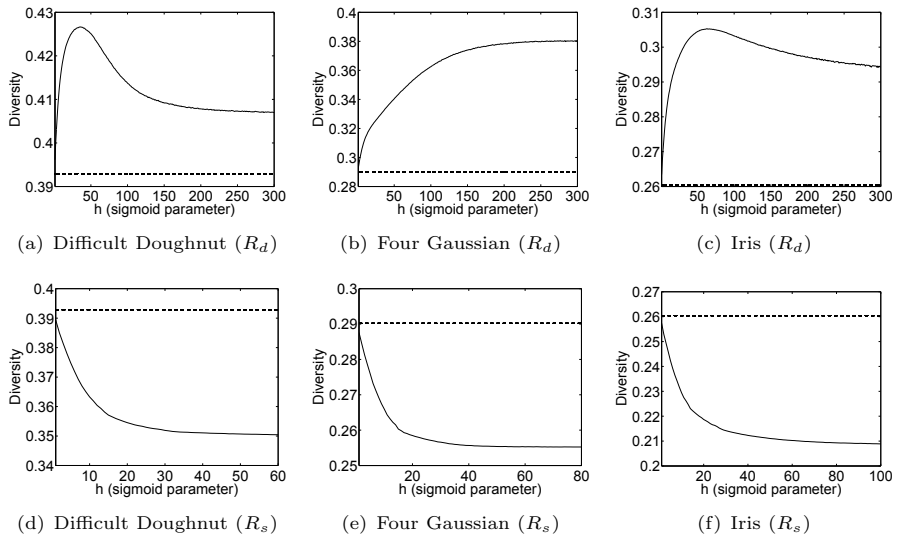
k	ANMI		Time [s]	
	MCLA rep.	Max rep.	MCLA rep.	Max rep.
2	0.746	0.751	0.037	0.941
3	0.747	0.763	0.069	1.054
4	0.922	0.922	0.046	1.174
5	0.896	0.883	0.047	1.295
6	0.857	0.848	0.056	1.420
7	0.813	0.825	0.062	1.550
8	0.794	0.804	0.062	1.676
9	0.784	0.798	0.072	1.767
10	0.773	0.793	0.079	1.883
11	0.778	0.794	0.078	2.047
12	0.796	0.796	0.085	2.181
13	0.794	0.798	0.092	2.290
14	0.804	0.806	0.097	2.377
15	0.806	0.812	0.103	2.524
16	0.814	0.818	0.108	2.638
17	0.819	0.823	0.112	2.754
18	0.825	0.829	0.120	2.844

methods to obtain a representative partition for a group, MCLA and Maximum ANMI, described in the previous section were evaluated. The results are shown in Table 1 for the Four Gaussian dataset. The first column indicates the k values considered: $[2, 18]$. This means that the ensemble contains 17 groups of partitions ($M = 17$). All group sizes are the same with $N_i = 20$. The second and third columns indicate the ANMI values obtained by the MCLA and the Maximum ANMI methods. The last two ones are the average elapsed time for both methods to get a representative partition, respectively. Although it is not shown in the table, the average of the ANMI of all group members against their own group was also measured. These values serve as a lower bound to measure the quality of the representatives. Both MCLA and Maximum ANMI produced a representative with an ANMI larger than this value. In Table 1 it can be seen that, although the representatives obtained by Maximum ANMI are the best in comparison to MCLA for almost all groups, their computation is much more intensive than MCLA. Similar results were observed for all the other datasets. This test, intended to measure the representative quality obtained by both methods, suggests that there seems to be no significant improvement to be worth the computation complexity. Therefore, the MCLA method has been chosen for obtaining the representative partitions.

4.2 Diversity Control

The diversity control method evaluation was carried out by firstly creating an original ensemble based on groups of partitions. Once the original ensemble was created, its diversity was measured. After that, a representative partition per group was obtained and all were compared, thus getting a relationship matrix. Once the representative weights were calculated, the diversity control method was employed by using the sigmoid function with an interval of values for h. One new ensemble per h value was generated, and its diversity measured.

The results of this experiment are shown in Figure 5. There are six figures, two per dataset, one using R_d and the other R_s. Each figure shows the diversity as a function of the sigmoid parameter h. The diversity of the original ensemble is indicated with dashed lines, and the solid ones show the diversity of the new



(a) Difficult Doughnut (R_d) (b) Four Gaussian (R_d) (c) Iris (R_d)

(d) Difficult Doughnut (R_s) (e) Four Gaussian (R_s) (f) Iris (R_s)

Fig. 5. Diversity control for three datasets (one per column), using R_d (first row) and R_s (second row).

ensembles. It can be seen that, while different patterns are observed in each dataset, the method produces a smooth diversity change in any case when R_d or R_s are used.

A closer look at the figures using R_d reveals that, although differently in each dataset, the method finds a point where the diversity is not incremented anymore. This could be explained by the fact that as the averages $\bar{r}_i$ are being contrasted, there is a value for h where the more diverse groups in the original ensemble are now similar to the rest of the new ensemble. This idea is reinforced by the fact that higher values for w_i produce groups with repeated partitions. It can be seen in Figures 5(a), 5(b) and 5(c) that, after the diversity control method reaches the maximum diversity, it finally converges at some value. In fact, as higher values for h are used, minor changes are observed in the sigmoid function, thus there is almost no differences in groups proportion among newly created ensembles. At this point, the only source of change are the group members randomly chosen from each group. For that reason, it is expected to find equally diverse ensembles at high values for h. On the other hand, when R_s is used (Figures 5(d), 5(e) and 5(f)), the diversity is monotonically decreased until it converges.

Once an ensemble is generated, the consensus partition derived from it can be compared against a reference partition of the dataset and get its accuracy. As it was previously mentioned, accuracy would be related to the ensemble diversity, what is directly dependant on the ensemble generation procedure. The proposed strategy to generate an ensemble based on groups of partitions performs well in comparison to other state of the art strategies. For example, the accuracy results obtained by our strategy is equal or better than those published in [10][4], where three ensemble generation strategies were used. For Four Gaussian, they

[4] Both measured with NMI.

obtained for MCLA an accuracy of 0.96, the same obtained here. For Iris, in contrast, they reached 0.81, while our strategy obtained an average of 0.88.

It is important to recall here something about how evaluation is generally carried out with cluster ensembles. As it was previously said in the introduction, studies in the area typically evaluate their new proposed methods by using the class labels that come with the dataset. As long as their results are more similar to the reference partition derived from these labels, more accurate they would be. Although this reference can represent a valid partition of the data, there could be no correspondence between the class labels and another equally valid structures found by a clustering algorithm. This is the unsupervised nature that is inherent in any clustering task. When a cluster ensemble framework is used, it is important to note that its components have different objectives, and it is sensible to evaluate them differently. Namely, the objective for the consensus function consists in maximizing the information shared between the consensus partition and the ensemble, as it was presented in (3). If this is not kept in mind, a low accuracy could be wrongly interpreted as a bad performing of the consensus function, or viceversa.

The other component of the cluster ensemble framework is the ensemble generation strategy. Although the ensemble diversity was found to be essential, it seems not to be clear enough how much diverse should an ensemble be, or even what diversity means. If the ensemble is good enough, the consensus function could obtain a better partition of the data. What a good ensemble is and what is the strategy to generate it seems to be related to its diversity, but it is still part of current research. In this work, we have presented a contribution to the area through a diversity control method which is able to precisely produce a set of ensembles with different diversities. This represents an advance to study the impact of the ensemble characteristics on the final consensus.

5 Conclusions

In this paper we have introduced a novel method to control the diversity of ensembles. It starts by creating an original ensemble based on groups of partitions, where its structure is appropriately used to estimate the relationship between each group. With this comparative information, the groups are weighted according to their impact on the ensemble diversity. By changing a parameter in the proposed method, it is possible to obtain ensembles with higher or lesser diversity. The empirical results suggest that this method is able to precisely control the diversity of ensembles, what represents a step toward a consistent approach to study the impact of diversity on the consensus partition. In addition, it has been found that the ensemble generation strategy based on groups of partitions produces more accurate results than classical strategies.

Future work includes an extensive study of another diversity measures like the non-pairwise ones, as well as some changes in the diversity control method to handle wider ranges of diversity. Besides, the individual study of the groups diversity could provide useful information to obtain better consensus partitions.

References

1. Domeniconi, C., Al-Razgan, M.: Weighted cluster ensembles: Methods and analysis. ACM Trans. Knowl. Discov. Data 2(4), 17:1–17:40 (2009)
2. Everitt, B.S., Landau, S., Leese, M.: Cluster Analysis. Wiley, 4th edn. (2009)
3. Fern, X.Z., Brodley, C.E.: Random projection for high dimensional data clustering: A cluster ensemble approach. In: ICML-2003. pp. 186–193 (2003)

4. Fred, A.L.N., Jain, A.K.: Combining multiple clusterings using evidence accumulation. IEEE Transaction on Pattern Analysis and Machine Intelligence 27, 835–850 (2005)
5. Ghosh, J., Strehl, A., Merugu, S.: A consensus framework for integrating distributed clusterings under limited knowledge sharing. In: In Proc. NSF Workshop on Next Generation Data Mining. pp. 99–108 (2002)
6. Gionis, A., Mannila, H., Tsaparas, P.: Clustering aggregation. In: In Proceedings of the 21st International Conference on Data Engineering (ICDE. pp. 341–352 (2005)
7. Hadjitodorov, S.T., Kuncheva, L.I., Todorova, L.P.: Moderate diversity for better cluster ensembles. Information Fusion 7(3), 264–275 (2006)
8. Hartigan, J.A., Wong, M.A.: Algorithm AS 136: A k-means clustering algorithm. Applied Statistics 28(1), 100–108 (1979)
9. Hubert, L., Arabie, P.: Comparing partitions. Journal of Classification 2(1), 193–218 (1985)
10. Iam-On, N., Boongoen, T., Garrett, S., Price, C.: A link-based approach to the cluster ensemble problem. Pattern Analysis and Machine Intelligence, IEEE Transactions on 33(12), 2396–2409 (2011)
11. Jain, A.K.: Data clustering: 50 years beyond k-means. Pattern Recogn. Lett. 31(8), 651–666 (Jun 2010)
12. Kleinberg, J.: An impossibility theorem for clustering. In: Neural Information Processing Systems. pp. 446–453. MIT Press (2002)
13. Kohonen, T.: Neurocomputing: foundations of research. chap. Self-organized formation of topologically correct feature maps, pp. 509–521. MIT Press, Cambridge, MA, USA (1988)
14. Kuncheva, L.I., Vetrov, D.P.: Evaluation of stability of k-Means cluster ensembles with respect to random initialization. Pattern Analysis and Machine Intelligence, IEEE Transactions on 28, 1798–1808 (2006)
15. Limin, L., Xiaoping, F.: A new selective clustering ensemble algorithm. In: e-Business Engineering (ICEBE), 2012 IEEE Ninth International Conference on. pp. 45–49 (2012)
16. Okabe, M., Yamada, S.: Clustering by learning constraints priorities. In: Data Mining (ICDM), 2012 IEEE 12th International Conference on. pp. 1050–1055 (2012)
17. Rashedi, E., Mirzaei, A.: A hierarchical clusterer ensemble method based on boosting theory. Knowledge-Based Systems 45(0), 83 – 93 (2013)
18. Strehl, A., Ghosh, J., Cardie, C.: Cluster ensembles - a knowledge reuse framework for combining multiple partitions. Journal of Machine Learning Research 3, 583–617 (2002)
19. Topchy, A., Jain, A., Punch, W.: Clustering ensembles: models of consensus and weak partitions. Pattern Analysis and Machine Intelligence, IEEE Transactions on 27(12), 1866–1881 (2005)
20. Xu, R., Wunsch, D.: Clustering. Wiley-IEEE Press (2009)
21. Yan, D., Chen, A., Jordan, M.I.: Cluster forests. Computational Statistics & Data Analysis (In press, available online) (2013), doi: 10.1016/j.csda.2013.04.010
22. Yi, J., Yang, T., Jin, R., Jain, A., Mahdavi, M.: Robust ensemble clustering by matrix completion. In: Data Mining (ICDM), 2012 IEEE 12th International Conference on. pp. 1176–1181 (2012)
23. Yu, Z., Wong, H.S., Wang, H.: Graph-based consensus clustering for class discovery from gene expression data. Bioinformatics 23(21), 2888–2896 (2007)